Microsoft

Understanding JavaScript and Coding Essentials

4.1. Manage and maintain JavaScript.
4.2. Update the UI by using JavaScript.

Agenda

1	JavaScript
2	Functions and Variables
3	jQuery and Third-Party Libraries
4	Locating and Accessing Elements
5	Listening and Responding to Events
6	Updating the Content of Elements
7	Adding Elements

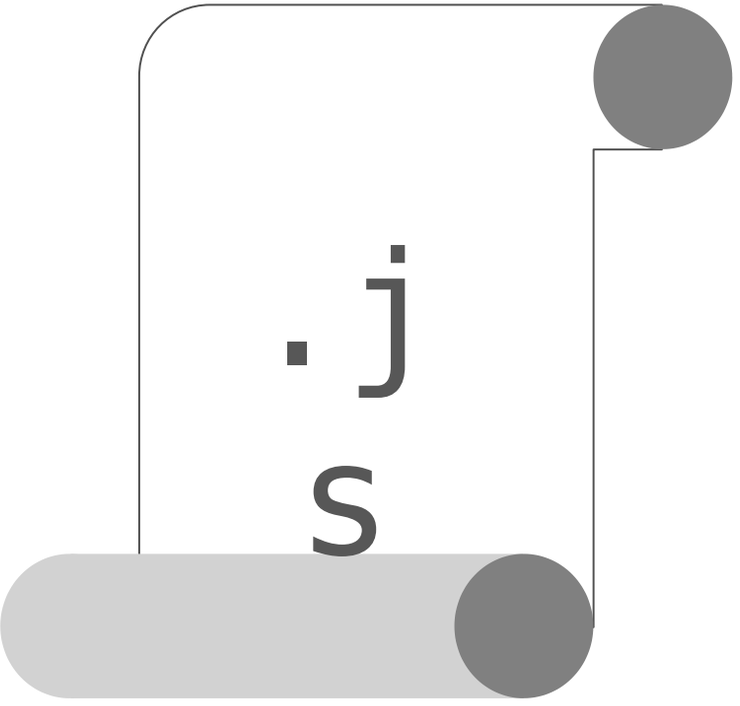
JavaScript

Building Interactive Applications

- **HTML5** and **CSS3** are awesome for creating beautiful websites
- However, today's users want an interactive Web experience
  - **Interactivity** allows a user to take an action and receive a response
- Implementing interactivity requires a programming language, such as **JavaScript**

What is JavaScript?

- **JavaScript** is a loosely-typed scripting language that is interpreted by a browser
- It changes how elements in an HTML document act and respond to user input
- We create **scripts** with JavaScript
  - Scripts are step-by-step instructions that tell a browser how to respond to events, such as a user clicking a button
  - The file extension for a script is .js



Connecting JavaScript with HTML

We can connect JavaScript to HTML documents in a couple of ways:

1. Embedding it with the <script> tag
2. Linking a separate JavaScript file to the HTML document

1

```
<script type="text/javascript">
   document.write("Hello World Wide Web");
</script>
```

2

```
<head>
 <script type="text/javascript"
src="Script.js"></script>
</head>
```

JavaScript Demo

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <meta charset="utf-8" />
    <title></title>
  </head>
  <body>
    <h1>This is a boring website!</h1>
    <script type="text/javascript">
1       document.write("Hello, World!");
    </script>
  </body>
</html>
```

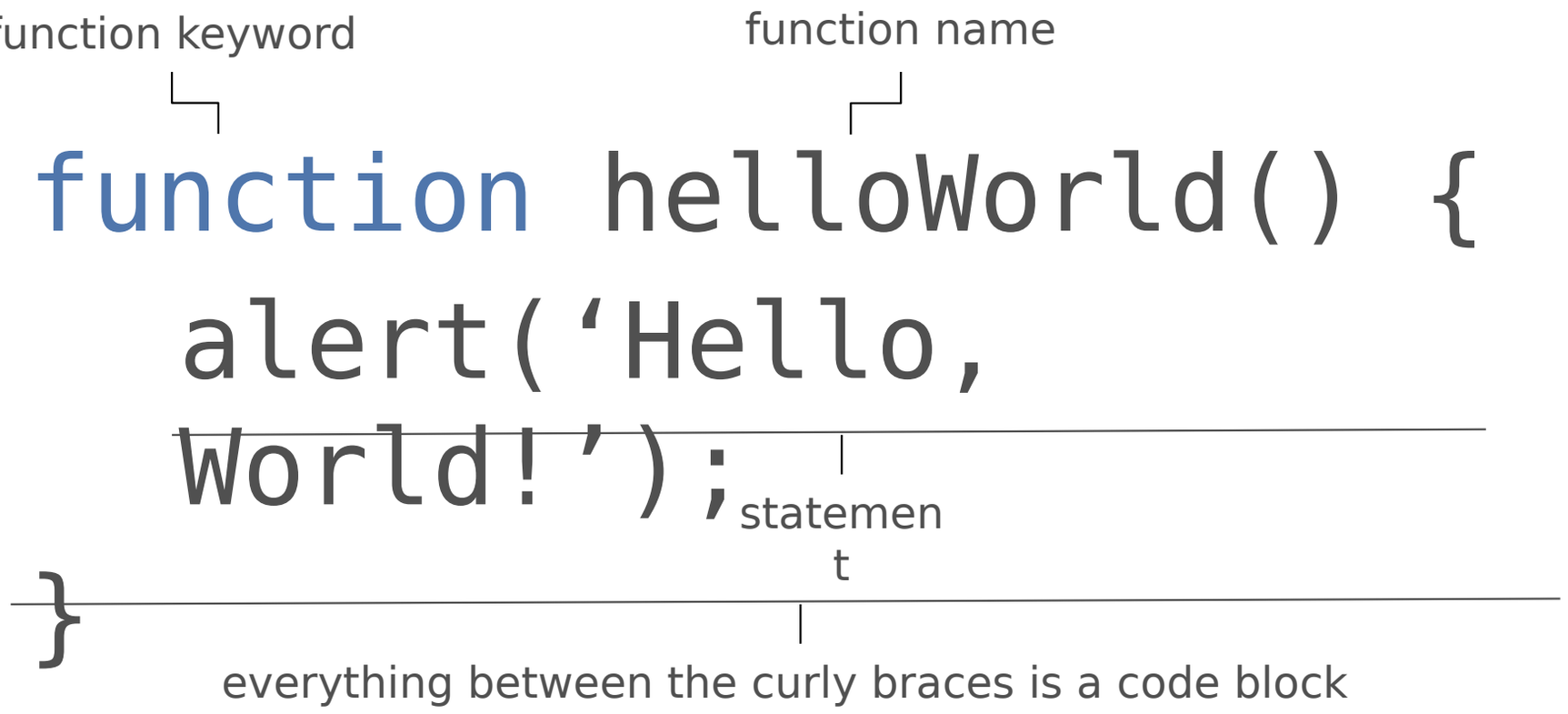
Functions and Variables

Functions

- A **function** is a group of statements that are combined to perform a specific task
- A **statement** is a line of code that performs an action
  - Statements should end with a semicolon (;)
- If different parts of a script repeat the same task, then you can reuse a function instead of repeating the same statements

```
function doSomethingAwesome() {
  var name = prompt(“What is your name?”);
  alert(name + “, you just did something
  awesome!");
}
```

How to Define a Function



Naming Functions

A function can have any name, but there are a couple guidelines that must be considered:

1. Don’t use any of the reserved words defined by JavaScript standards.
   - See the full list of JavaScript reserved words by clicking here
2. The name must be made of letters, digits, underscores, or dollar signs
   - It can’t start with a number though!

Definition and Execution of Functions

- The way that a function is defined is different from how it is executed by a browser
- A function's definition outlines its name, any parameters it may take, and its statements
  - **NOTE:** a definition doesn't perform any of a function's statements
- When a function is called, the browser will execute all of the statements within the function

Defining the Function

```
function doSomethingAwesome() {
    var name = prompt("What is your name?");
    alert(name + ", you just did something
    awesome!");
}
```

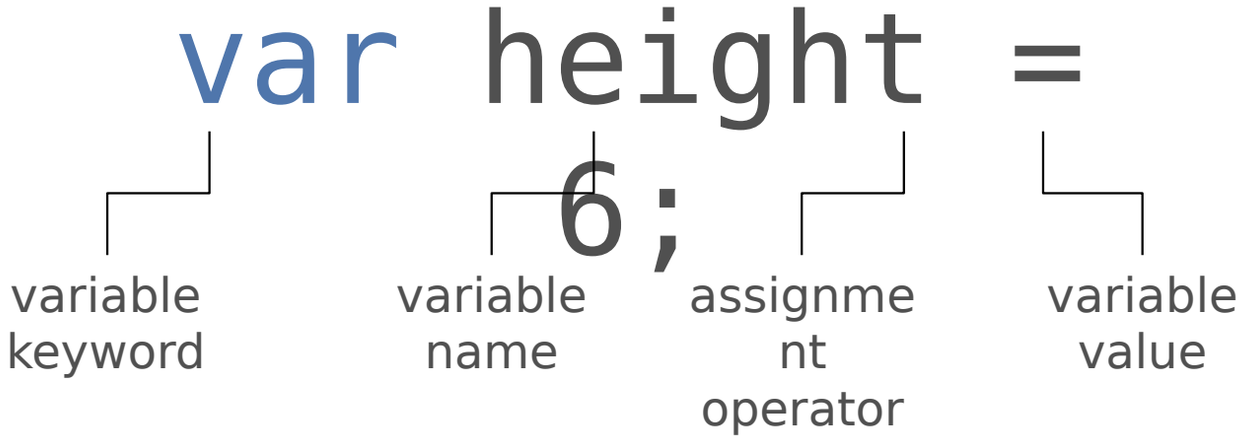
Calling the Function

```
<input type="button" value="Click Me"
    onclick="doSomethingAwesome()">
```

Variables

- Scripts have to temporarily store pieces of information
- These bits of data can be stored as **variables**
  - It’s called a variable because its values can vary every time a program is run
- Variables can be defined using the `var` syntax with a unique keyword, such as `height` or `width`

How to Declare a Variable



Rules for Naming Variables

Variable names must start with a letter, dollar sign ($), or an underscore (_).

It must NOT start with a number.

1

Variable names can contain letters, numbers, dollar signs, and underscores, but NOT dashes (-) or periods (.).

2

You cannot use keywords or reserved words.

3

Variables are case sensitive, which means that `thisVariable` is different from `ThisVariable`.

4

Use names that describe the information you are storing.

5

If a variable name uses two or more words, capitalize the first letter of ever word AFTER the first word.

6

Types of Data

- Numbers `1, 2, 3`
- Strings `‘Zombies freak me out!’`
  - Must always be surrounded by quote marks
- Boolean `true, false`

Comments

Add comments to your script to explain what it does

- It will also make your code easier for others to read and understand

Add a single-line comment by placing two forward slash characters // in front of your comment

- Anything after the slashes won't be interpreted by the browser

Add a multi-line comment by starting with the /* characters and ending with the */ characters

- Anything between these characters won't be interpreted by the browser

JavaScript

```
/*These comments are typically
reserved for describing how an entire
script file works or to comment out
an entire block of script. */

//this function does something
awesome!
function doSomethingAwesome() {
    var name = prompt("What is your
    name?");
    alert(name + ", you just did
    something awesome!");
}
```

jQuery and Third-Party LIbraries

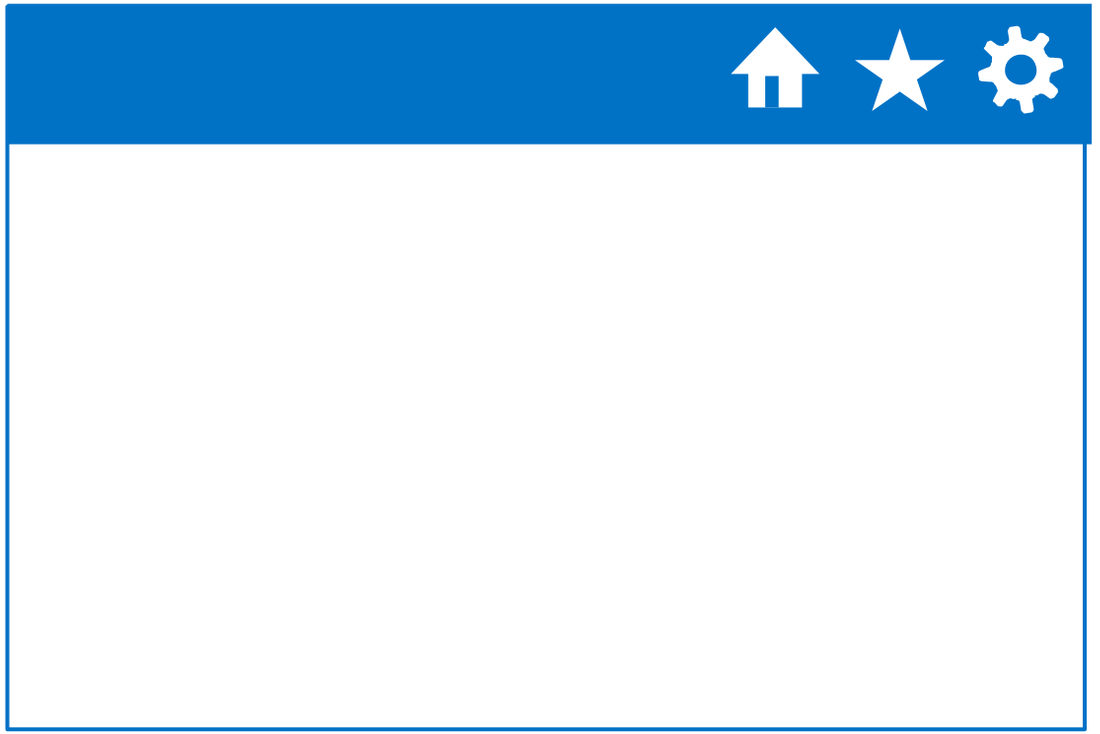
JavaScript Libraries

- JavaScript Libraries are made of code that other programmers have already developed
  - libraries include pre-written functions and statements that you can use to create programs
- Use a library by linking its file to your web page
- One of the most popular JavaScript libraries is **jQuery**
  - jQuery allows you to use CSS-like selectors and its methods to perform functions with minimal code

Locating and Accessing Elements

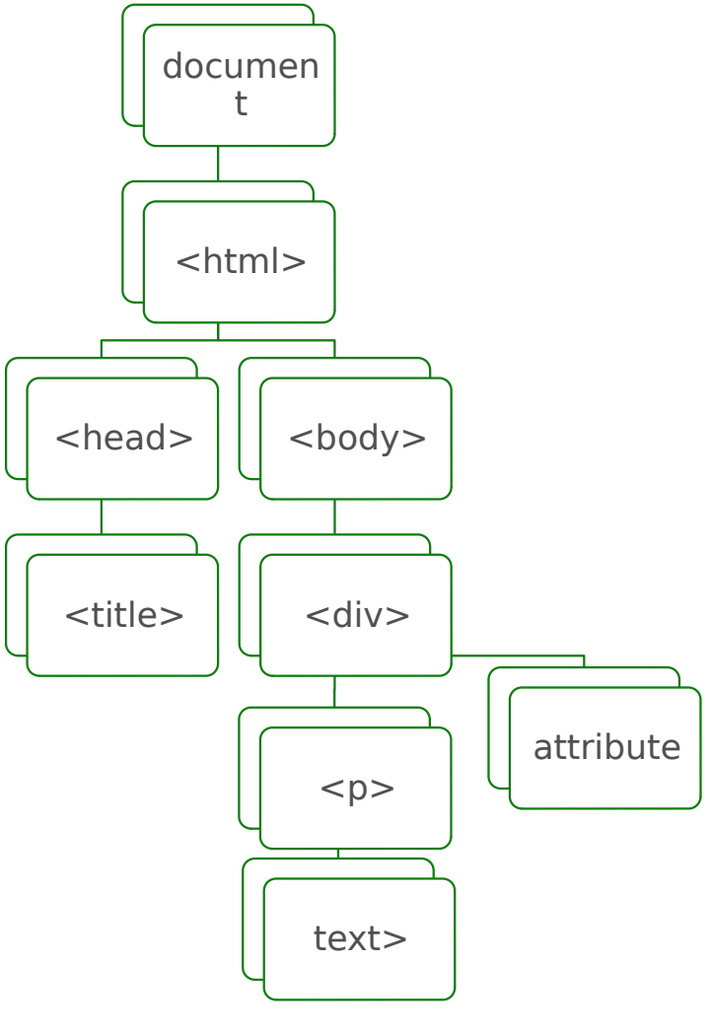
Objects in JavaScript

- An HTML element is an object, similar to a house or a car
- Just as with real life objects, we can access and modify HTML objects that appear on a screen
- The creation of interactive Web pages and apps relies on our ability to manipulate **objects** on a screen
  - Objects are models of things in the real world that were built using data
- Objects are grouped into



Document Object Model (DOM)

- The Document Object Model (DOM) creates a model of a Web page
- The DOM is used to update content, structure, and styles on the fly
- The topmost object is the **document object**, which represents the page as a whole
  - It has child objects that represent individual elements on a page



Locating and Accessing Elements

- We can access objects in the DOM using an element's ID
- To do so, we can use the document object's `getElementById()` method
  - This means that the element must have an ID
- Using this method allows you to manipulate the contents of that element

```
document.getElementById('demo');
```

object

method name

parameter

getElementById() Demo

```
<body>

  <h1>Get today's date and add it to an element on the
page.</h1>

  <p id="demo"></p>

  <script type="text/javascript">
   document.getElementById("demo").innerHTML=Date();
  </script>

</body>
```

Listening and Responding to Events

Events in Programming

- Events are actions that a user takes
- JavaScript features **event handlers**, which respond to specific user events
  - For example, the `onClick` event handler responds to clicks on screen
- Event handlers respond by executing functions

Event Handlers	Associated Events
`onsubmit`	form submission

`onkeydown`
`onkeypress`
`onkeyup`
	keystrokes

`onclick`
`onmousedown`
`onmouseup`
	mouse or touchpad clicks

`onload` `onunload`	page loading/unloading
`onselect`	item selection

Event Handlers Demo

```
<body>
    <p>Select some of the text:
     <input type="text" value="Hello, World!"
onselect="myFunction()"></p>
    <p id="demo"></p>

    <script>
        function myFunction() {
            document.getElementById('demo').innerHTML =
     "Selecting text is awesome!";
        }
    </script>
</body>
```

Updating the Content of Elements

Updating Content in Elements

- Use the `innerHTML` property to change content or insert new content between element tags
  - It can be used on any element
- To change content, set the `innerHTML` property to the desired string
  - To do this, we must use the equals symbol (=)
- To remove content, set it to an empty string

innerHTML Demo

```
<body>

  <h1>Updating Content</h1>
  <p id="demo"></p>

  <script type="text/javascript">
   document.getElementById("demo").innerHTML=‘Using
  JavaScript is super fun!’;
  </script>

</body>
```

Adding Elements

The createElement method

- Make elements, like images, appear on screen with the document object's createElement method
- Add the element to the screen using the appendChild() method

JavaScript

```
function show_image(src, width, height,
alt) {
    var img =
document.createElement("img");
    img.src = src;
    img.width = width;
    img.height = height; img.alt = alt;
    // Adds it to the <body> tag
    document.body.appendChild(img);
}
```

HTML

```
<button
    onclick="show_image ('dog.jpg',
    276,110, 'Stella');">
    Display an image!
</button>
```

createElement Demo

```
<body>

  <h1>Creating Elements</h1>
  <p id="demo"></p>

  <button onclick="show_image
('dog.jpg',300,400,'Stella');">Dis
play an image!
  </button>

</body>
```

```
<script>
function show_image(src, width, height,
alt) {
    var img =
document.createElement("img");
    img.src = src;
    img.width = width;
    img.height = height;
    img.alt = alt;
    // Adds it to the <body> tag
    document.body.appendChild(img);
}
</script>
```

Summary

1	JavaScript
2	Functions and Variables
3	jQuery and Third-Party Libraries
4	Locating and Accessing Elements
5	Listening and Responding to Events
6	Updating the Content of Elements
7	Adding Elements